

IN THE CLAIMS

1. (Currently Amended) A system that allows analysis of software running in a tamper- resistant environment, the system comprising:

a processor which monitors at least one instance of software execution, wherein the one instance is identified and selected by an end-user to be monitored by the processor, wherein the end-user is a user that initiates execution of the software at a system associated with the end-user, and wherein the processor creates a log entry with at least one set of data derived from the one instance of software execution in response to the one instance being identified and selected to be monitored, whereby the set of data is used to diagnose the software execution;

an encryption system which generates at least one symmetric key and encrypts the log entry for the at least one set of data using the symmetric key, wherein the encryption system encrypts the symmetric key using a public key associated with the encryption system;

a log file of a relatively-fixed size which stores the log entry for the at least one set of data which has been encrypted, and wherein the log file includes the symmetric key which has been encrypted with the public key;

random data in the log file when it is originally created and which is replaced by log entries so that a size of the log file including log entries appears to be a substantially constant size; and

a pointer which identifies a next storage location for a next log entry so that a last log entry can be determined and the next log entry can be positioned in a location in the log file after a previous log entry.

2. (Previously Presented) A system including the elements of Claim 1 wherein the system includes a transmission system for sending the log file, upon command, to a secure processing location away from the system in which the log file was created.

3. (Previously Presented) A system including the elements of Claim 1 wherein the system includes a system for wrapping around and filling the log file from a beginning when the log file has been filled, allowing the log file to remain at a substantially-constant size even after the log file has been filled with data and a new entry is received.

4. (Previously Presented) A system including the elements of Claim 1 wherein the system includes a mechanism for obscuring the log entry which has been created.

5. (Previously Presented) A system including the elements of Claim 4 wherein the mechanism for obscuring the log entry which has been created includes a printing function for writing into the log file.

6. (Previously Presented) A system including the elements of Claim 2 wherein the system includes a mechanism for receiving an indication from a user that transmission is desired and transmits the log file in response to that indication.

7. (Currently Amended) A system including the elements of Claim 1 wherein the system further includes a mechanism for receiving an input from an end-user that initiates logging of log entries into the log file each time logging is desired by the user.

8. (Previously Presented) A system including the elements of Claim 1 wherein the system further includes an initializing mechanism for determining each instance logging is to begin and initiating logging of log entries only in response to that initializing mechanism.

9. (Previously Presented) A system including the elements of Claim 1 wherein the system uses the public key to encrypt the log entry which has been created and a private key corresponding to the public key is used to decrypt the log entry which has been created at a secure location.

10. (Currently Amended) A method for diagnosing software in a tamper-resistant environment comprising the steps of:

- generating a log file full of random data, wherein the log file is of a substantially constant size;

- generating at least one symmetric key, wherein the symmetric key resides within the log file;

- turning on logging and establishing a pointer for a location of a next logged software operation activity;

- monitoring at least one software operation activity within the tamper-resistant environment and generating messages in response to at least one instance of software execution within the tamper-resistant environment, wherein the software operation activity is identified and selected by an end-user to be monitored, wherein the end-user is a user that initiates execution of the software at a system associated with the end-user;

- encrypting, in response to the monitoring, a record associated with each generated message using the symmetric key;

- encrypting using the symmetric key, wherein ~~[[the]]~~^a encryption system encrypts the symmetric key using a public key associated with the encryption system;

- logging into a log entry into a log file, at least one software operation activity relating to a generated message by replacing random data with an encrypted record of the software operation activity;

- placing into the log file the symmetric key which has been encrypted with the public key;

- moving the pointer when the log entry has been made to a next available log position;

- wrapping the pointer to a beginning of the log file when the log file is full of log entries;

- sending the log file to a secure location where the log file can be decrypted and analyzed;

and

- analyzing decrypted log file data and providing information for diagnosing software in the tamper-resistant environment.

11. (Previously Presented) A method including the steps of Claim 10 wherein the step of turning on logging includes the steps of receiving a user input indicating that logging is desired and initiating the logging in response thereto.

12. (Previously Presented) A method including the steps of Claim 10 wherein the step of at least one software operation activity further includes the steps of determining whether the software operation activity is to be logged, and if so, determining when to encrypt the software operation activity to obscure what is being logged.

13. (Previously Presented) A method including the steps of Claim 10 wherein the step of logging the software operation activity further includes the steps of determining a next available log position, replacing existing data in the next available log position with data from the software operation activity, and updating the pointer to provide a location of next logged software operation activity.

14. (Previously Presented) A method including the steps of Claim 10 and further including the step of receiving a command from a user that indicates that sending the log file to a remote location is desired and transmitting the log file in response thereto.

15. (Currently Amended) A method of analyzing the operation of software in a remote protected processing environment, the method including:

receiving from the remote protected processing environment an encrypted log file of substantially-constant size comprising at least one log entry with at least one set of data derived from at least one instance of software execution monitored in response to an end-user identifying and selecting the one instance of software execution, wherein the end-user is a user that initiates execution of the software at a system associated with the end-user, whereby the set of data is used to diagnose the software execution, wherein the set of data within the encrypted log file has been encrypted with at least one symmetric key included within the encrypted log file, and wherein the symmetric key has been encrypted by a public key associated with an encryption system;

determining a decrypting key for the encrypted log file and decrypting the encrypted log file;

determining a private decrypting key corresponding to the public key associated with the system;

analyzing, using the decrypting key and the private decrypting key, the log entry at the remote protected processing environment to determine whether an operation of the remote protected processing environment corresponding to the at least one set of data derived from at least one instance of software execution is appropriate; and

reporting results of the analyzing step.

16. (Currently Amended) A method providing the steps of Claim 15 and further including providing an instruction to initiate a logging of messages each time logging is desired by the end-user and an instruction to send to the encrypted log file to a remote system for analysis.

17. (Previously Presented) A method providing the steps of Claim 16 wherein the instruction to initiate logging of messages includes the step of initiating programming within the remote protected processing environment to replace information in the encrypted log file with encrypted information relating to the operation of the remote protected processing environment.

18. (Previously Presented) A method providing the steps of Claim 17 wherein the step of replacing information in the encrypted log file includes the step of replacing random data which was placed in the encrypted log file when it was created.

19. (Previously Presented) A method providing the steps of Claim 17 wherein the step of replacing information in the encrypted log file includes the step of using a pointer to a next location in the encrypted log file and the pointer wraps to a beginning of the log file after the encrypted log file has been filled.

20. (Currently Amended) A computer readable storage medium for analyzing software running in a tamper-resistant environment, the computer readable storage medium comprising instructions for:

recording at least one set of data serviced from at least one instance of software execution identified and selected by an end-user to be monitored whereby the set of data is used to diagnose the software execution wherein the end-user is a user that initiates execution of the software at a system associated with the end-user;

generating at least one symmetric key;

encrypting the symmetric key using a public key associated a client computer system

encrypting the recording of the at least one set of data using the symmetric key;

recording the at least one set of data, which has been encrypted sequentially in a storage block of a substantially fixed size, wherein the storage block includes the symmetric key which has been encrypted with the public key;

maintaining a pointer to a next available location for recording the at least one set of data sequentially in the storage block;

responding to a command and sending an encrypted log file comprising the at least one set of data which has been encrypted and sequentially recoded in the storage block to a remote location for decryption and analysis.

21. (Previously Presented) The computer readable storage medium of claim 20, further comprising instructions for:

initializing the storage block of a substantially fixed size with random information which has been encrypted to provide a block of apparent data.

22. (Previously Presented) The computer readable storage medium of claim 20, further comprising instructions for:

writing the at least one set of data which has been encrypted and recorded in a sequential order in the storage block of the substantially fixed size and for wrapping around when an end of the storage block of the substantially fixed size is reached.